



混合流媒体系统的资源搜索机制研究^{*}

郭大纲¹, 卓明琴², 张继荣³

(1. 西安邮电学院邮电技术公司 西安 710061;

2. 中兴通讯股份有限公司南京研发中心 南京 210012;

3. 西安邮电大学通信与信息工程学院 西安 710121)

摘要: 在基于内容分发网络和对等网络的混合流媒体系统中引入基于资源知名度的混合搜索机制, 依据不同的资源知名度, 分别采用结构化搜索和泛洪搜索, 提高了搜索效率。结构化部分采用改进型 Chord 搜索机制, 其路由时延较传统 Chord 搜索大幅度降低; 泛洪部分采用基于布鲁曼滤波器的泛洪搜索, 相比以往的泛洪搜索, 网络中的冗余查询消息明显减少。此外, 系统还根据资源知名度的不同分别采用多关键字查找和语义相关度匹配来支持资源的模糊查找, 提高了搜索成功率。

关键词: 资源知名度; 结构化搜索; 泛洪搜索; 布鲁曼滤波器; 模糊查找

doi: 10.3969/j.issn.1000-0801.2014.02.004

Research of Resources Search Mechanism on Hybrid Stream Media System

Guo Dagang¹, Zhuo Mingqin², Zhang Jirong³

(1. Posts and Telecommunications Technology Company,

Xi'an University of Posts & Telecommunications, Xi'an 710061, China;

2. Nanjing R&D Center, ZTE Corporation, Nanjing 210012, China;

3. Department of Communication & Information Engineering, Xi'an University of
Posts & Telecommunications, Xi'an 710121, China)

Abstract: A hybrid resources search mechanism based on the popularity of resources was proposed in the hybrid peer to peer stream media system based on CDN and P2P. A different search mechanism was used according to the different popularity of resources, such as structured search or flooding. By this way, the efficient of search was effectively improved. In the part of structured search, an improved Chord search mechanism was proposed. Compared with the traditional Chord search mechanism, the router delay was greatly reduced. A flooding search mechanism based on bloom filter was proposed in the part of flooding. Compared with the traditional flooding search mechanism, it can significantly lower the redundancy transmission of search request. Also, according to the different popularity of resources, multi-keyword query and the concept of semantic correlation were proposed to support fuzzy search, and the rate of successful search was greatly increased.

Key words: resources popularity, structured search, flooding search, bloom filter, fuzzy search

^{*} 工业和信息化部通信软科学项目(No.2013R45)



1 引言

传统的基于 P2P(peer to peer,对等)网络的流媒体系统中,资源的搜索方法是单一的基于结构化搜索或非结构化搜索,并未将这两种模式进行优势互补。事实上,针对不同的资源分布,两种搜索模式体现出不同的特性:结构化搜索模式能准确命中资源,且具有相对确定的路由跳数和响应时间^[1],其缺点在于为了保持网络逻辑拓扑而引入的动态维护开销过大且过程较复杂,增大了网络节点的处理负荷;非结构化搜索模式消除了对逻辑拓扑的依赖所带来的维护开销,并且对知名资源具有很好的搜索效率与质量^[2],但其对稀有资源的查找效率不高,且一般响应时间较长。本文研究的混合流媒体系统^[3]综合考虑了这两种搜索模式的优势,并进行了改进。根据搜索资源的知名度采取不同的搜索方法,即当搜索稀有资源时采用结构化搜索机制,当搜索热门资源时则采用非结构化搜索机制,从而大大提高了搜索效率。为了更好地提高搜索命中率,还根据不同的资源知名度,采取不同的策略支持模糊查找。主流模糊查找大体分为 2 类,即多关键词搜索(multi-keyword query,MKQ)和语义搜索。

在处理稀有资源时,使用结构化搜索过程,结构化 P2P 网络中散列函数的随机性和均匀分布性导致它不能很好地支持语义搜索,而稀有资源在网络中的数量较少,采用 MKQ 所引入的分割发布对发布节点的存储空间和网络带宽占用量很小,且实现方法简单,因此在结构化搜索部分,使用 MKQ 支持模糊查找。

在处理知名资源时,使用非结构化搜索过程,由于资源词条较为热门,在网络中的存储量很大,采用 MKQ 将会大量占用发布节点的存储空间和网络带宽,因此 MKQ 在此处不适用,而非结构化 P2P 网络能够很好地支持语义搜索,研究发现,利用语义相关度可以较精确地在众多相似资源词条中找到与要求最接近的资源词条,因此在非结构化部分,采用语义相关度来支持模糊查找。

2 混合流媒体系统结构及原理

2.1 混合流媒体系统结构

在混合流媒体系统中,将媒体分发网络分为 2 层,分别为骨干网层和边缘网层。在骨干网层次,采用 CDN 技术,通过在网络边缘合理布置代理缓存服务器(边缘服务器),将视频内容推送到靠近用户端侧;在边缘网层次,以不同的代理缓存服务器为中心,组成多个自治的、相互独立的 P2P 流媒体网络,利用 P2P 技术进行流媒体的共享,具体系统结构如图 1 所示。

系统的整体设计结合了互联网的结构特点,流媒体中心服务器(central server,CS)、请求路由服务器(request routing server,RRS)和分布在各自自治系统的边缘服务器(edge server,ES)通过骨干网互连组成流媒体 CDN,各互联网自治系统内的边缘服务器与客户机组成与其他自治系统相互独立的混合流媒体网络。边缘服务器是混合流媒体网络中的超级节点(super peer,SP),客户机是叶子节点(leaf peer,LP)。

2.2 系统工作原理

在上层应用中,中心流媒体服务器负责在初始状态发

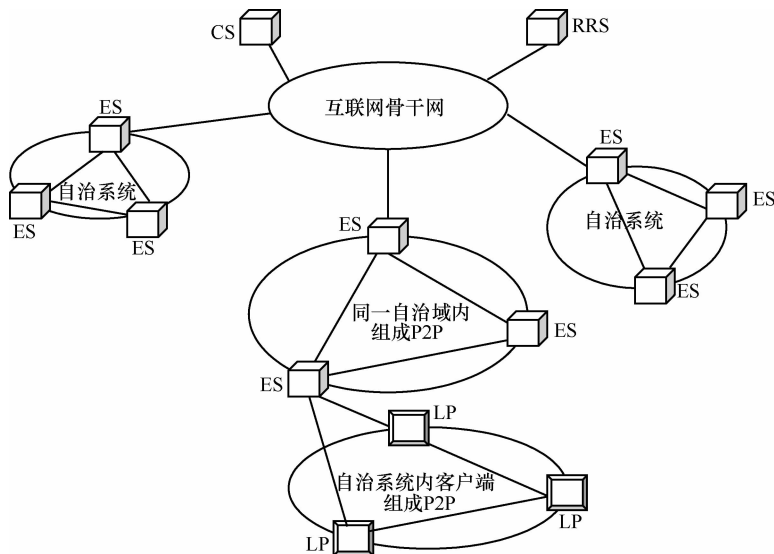


图 1 混合流媒体系统结构

布流媒体资源。RRS 负责对新加入的 LP 进行身份鉴权,若鉴权成功,则根据 LP 的 IP 地址或端口号通过 SHA-1 算法为其分配一个全局唯一的 ID,并向 LP 指定一个对它最合适的 SP。RRS 还负责在资源知名度更新过程中对 CSP (chief super peer) 进行选择。

LP 按照一定的词条抽取规则对本节点共享的资源进行词条抽取后,重定向到 SP,并向 SP 传送其共享资源的词条,供 SP 进行本地资源的统计和更新。

SP 根据全局资源变化率 η 计算出资源在网络中的实时数量,并与系统预设的泛洪阈值 r 进行比较,将资源分别归入稀有资源集和热门资源集中。为了支持模糊查找,SP 还需向 LP 发送实时的 η 值,LP 根据 η 值判断其存储资源的流行度,并对其中的稀有资源进行分割和发布,以支持 MKQ;对于热门资源,直接以二元组(词条、节点 IP 地址)的形式存储于本节点中。

在需要搜索某个流媒体资源时,LP 会向 SP 发出搜索请求,SP 根据 LP 输入的关键码 Key 判断其在网络中的流行度,即 SP 在稀有资源集和热门资源集中查找含有 Key 的资源词条,按照式(1)和式(2)计算两个集中相关词条与 Key 的平均语义相关度 SR_{aver} 。

$$SR(Key, S) = \left[\frac{1}{n} \sum_{i=1}^n [(1 - |p_i - q_j|) \wedge SR(Key_i, S_j)]^p \right]^{\frac{1}{p}} \quad (1)$$

$$SR_{aver} = \frac{\sum_i SR(Key_i, S)}{n}, 1 < i < n \quad (2)$$

其中, n 为相关词条个数, p_i, q_j 为 Key 和 S 相关关键词的隶属度, $p \geq 1, p_i \geq 0, q_j \leq 1, SR(Key_i, S_j)$ 根据知网义原预先设定^[4]。

通过计算,SP 优先选择 SR_{aver} 较大的集合作为候选集,若热门资源集中相关词条与 Key 的平均语义相关度大些,则 SP 选定热门资源集为候选集,并告知 LP 采用改进的泛洪机制和语义相关度匹配进行模糊搜索,在搜索结束后,将结果按照一定的规则显示给 LP,若 LP 不满意,则重新以 DHT(distributed hash table)机制和 MKQ 结合进行模糊搜索。若在全球搜索完成后仍然没有满意结果,SP 会向上层 CS 请求该资源。

2.3 主要节点的数据结构定义

2.3.1 SP 中的主要数据结构和数据成员

SP 中的主要数据结构和成员如下:邻居 SP 表、本地

资源表、本地节点的拓扑表、泛洪阈值、本地网络规模、全局拓扑维护周期。

2.3.2 LP 中的主要数据结构

在 LP 中,对传统的 Chord 网络节点路由表(finger table)进行改进,增加了逆向路由。双向路由表(逆时针路由表,前继节点,后继节点,顺时针路由表)见表 1 和表 2,其中, $i = \log_3 N, N$ 为网络规模。其他主要数据结构介绍如下。

表 1 顺时针路由

序号	开始	节点间隔	节点编号
1	$ID + 3^0 \bmod N$	$ID + 3^0 \bmod N \sim ID + 2 \times 3^0 \bmod N$	ID_i
2	$ID + 2 \times 3^0 \bmod N$	$ID + 2 \times 3^0 \bmod N \sim ID + 3^1 \bmod N$	ID_j
...	...	...	...
$i-1$	$ID + 3^{i/2-1} \bmod N$	$ID + 3^{i/2-1} \bmod N \sim ID + 2 \times 3^{i/2-1} \bmod N$	ID_k
i	$ID + 2 \times 3^{i/2-1} \bmod N$	$ID + 2 \times 3^{i/2-1} \bmod N \sim ID + 3^{i/2} \bmod N$	ID_m

表 2 逆时针路由

序号	开始	节点间隔	节点编号
1	$ID - 3^0 \bmod N$	$ID - 3^0 \bmod N \sim ID$	ID_a
2	$ID - 2 \times 3^0 \bmod N$	$ID - 2 \times 3^0 \bmod N \sim ID - 3^0 \bmod N$	ID_b
...	...	...	...
$i-1$	$ID - 3^{i/2-1} \bmod N$	$ID - 3^{i/2-1} \bmod N \sim ID - 2 \times 3^{i/2-2} \bmod N$	ID_c
i	$ID - 2 \times 3^{i/2-1} \bmod N$	$ID - 2 \times 3^{i/2-1} \bmod N \sim ID - 3^{i/2} \bmod N$	ID_d

- 本节点资源表(资源词条列表),邻居表。由本地 LP 代为发布、存储 LP 代为发布的资源及其源节点信息,此时 LP 被称为发布节点 Distr_ID。
- 缓存表(cache table):存储之前网内成功搜索的资源信息。由于在 Distr_ID 退出时,它代为发布的资源就会被转存到它的后继节点,因此在缓存表中将存储它的若干个后继节点,可以确保在其退出时,转而访问它的后继节点,进而找到资源的源节点 S_ID。

3 改进的 Chord 搜索机制

混合流媒体系统中的结构化搜索(DHT 机制)部分采用改进的 Chord 搜索机制,节点路由表与以往的 Chord 网络节点路由表不同:路由表节点间隔以 3 为基数;增加了



逆向路由表,查找以双向的方式进行;增加了缓存表,若之前查找过相同的资源,便可以快速地再次定位;以 MKQ 的查找方法来支持 DHT 部分的模糊查找。

3.1 节点加入(弱稳定算法)

假设 LP_i 的节点 ID 为 ID_i , 所属的 SP 记为 SP_i , 泛洪阈值为 r , 网络规模为 N 。

当有节点 LP_i 要加入网络时,通过 RRS 鉴权后重新定向到 SP_i , 向 SP_i 发送本节点资源词条信息。 SP_i 开始进行拓扑的弱稳定维护,即 SP_i 向 LP_i 发送它的后继节点 S_i 信息和实时的全局资源变化率 η , 则 LP_i 及其相关节点进行如下操作:

```

 $LP_i.succesor=S_i;$ 
 $LP_i.predecessor=S_i.predecessor;$ 
 $S_i.Predecessor.succesor=LP_i;$ 
 $LP_i.Predecessor=LP_i$ 

```

同时, LP_i 复制 S_i 的路由表信息。经过上述弱稳定维护后, LP_i 成功加入逻辑拓扑环中,并保持了 Chord 环的完整性,而其他相关的节点暂时不更新自己的路由表。

之后, LP_i 根据 η 进行资源发布: 若资源为稀有资源, 则将资源词条以较小单位 (如以单个汉字或字符为单位) 通过 SHA-1 算法得到资源标识 SID, SID 和节点标识 ID 属于同一个命名空间。 LP_i 根据 SID 的值将资源发布到与 SID 最接近的 ID 节点上。例如, 节点 LP_i 中存有“创世纪”这一稀有电影资源, 则 LP_i 将“创世纪”的资源属性按照抽取规则进行抽取后得到资源词条 (如片名、主演、电影情节和关键字等), 将词条内容进行分割, 如“创世纪”被分割成“创”、“世”、“纪”, 并分别通过散列函数发布到相应的节点 ID_n , ID_k 和 ID_m 上, 每个发布节点包含其所负责的关键码和对应的资源词条以及相应的节点信息。

此外, LP_i 还将通过探测邻居节点与其距离来初始化自己的邻居表。初始状态下, 节点的缓存表为空或保留上次在网内的成功查找记录。

3.2 查询路由

当 LP_i 需要搜索某个流媒体资源时, 例如搜索电影“创世纪”时, 用户只需输入影片的若干个词条关键码便可进行模糊查找, 具体查找过程如下。

(1) LP_i 首先查询本地缓存表, 若之前 LP_i 成功查询过该资源, 缓存表中会有相应的记录, LP_i 可以直接定位到资源发布节点 Distr_ID, 从而找到资源的源节点 S_ID。

(2) 若缓存表中无相关记录, LP_i 向所属 SP_i 发送查询请

求, SP_i 通过分析得出候选集, 若稀有资源集合中相关词条与输入请求码字的平均语义相关度较大, 则 SP_i 指示 LP_i 采用 DHT 搜索机制, 即采用改进的 Chord 机制和 MKQ 结合进行模糊查找。

(3) 若 LP_i 被指示采用 DHT 机制搜索, 它将输入的关键字分割成单位关键码并经过 SHA-1 算法得到资源标识, 如 LP_i 输入“世纪”时, 将其分割为“世”、“纪”, 并调用 SHA-1 算法分别得到资源标识 SID_0 和 SID_1 , LP_i 将按照 SID_0 或 SID_1 进行搜索。

假设 LP_i 以 SID_0 开始搜索, 首先判断 SID_0 与 $ID_i+N/2$ 的大小关系, 若 SID_0 小于 $ID_i+N/2$, 则在顺时针路由表中进行查找, 否则, 在逆时针路由表中进行查找; 查找时首先查看 $LP_i.succesor$ (后继节点) 的 ID, 若 SID_0 介于 LP_i 与 $LP_i.succesor$ 之间, 则资源存储在 $LP_i.succesor$ 上, 否则, 在路由表中找到小于 SID_0 但最接近它的节点 ID_j , 将查询请求转发给 ID_j , ID_j 节点重复 ID_i 节点的过程, 经过有限轮的转发和查找, 最终找到和 SID_0 最接近的节点 ID_k , 负责 SID_0 的节点 ID_k 将返回所有包含“世”字的资源列表, LP_i 在返回的结果中再根据关键码“纪”搜索词条中包含“纪”字的所有词条, 剔除不含“纪”字的资源词条, 如果还有其他的关键码则继续搜索。此过程可能找到精确的目标文件, 也可能只是一些符合条件的资源词条的集合, 用户自己再进行选择。知道的关键码越多, 找到的词条范围越小, 用户越容易做选择。

(4) 通过选择后, LP_i 将最终的结果集合 $\{LP_s, s \geq 1\}$ 按照匹配度进行排序, 并检查 $LP_n \in \{LP_s, s \geq 1\}$ 是否存在于自己的邻居表中。若存在, 则认为 LP_n 距离 LP_i 最近, 否则, LP_i 将计算与各个 LP_s 的距离。 LP_i 将优先选择与自身距离最近且资源匹配度较高的节点进行连接。

(5) 若 LP_i 不满意返回的结果, 则 LP_i 可激活以改进的泛洪机制和语义相关度进行模糊查找。当搜索完成后, 若仍没有满意结果, LP_i 将上报 SP, 由 SP 向上层 CS 请求资源。

(6) LP_i 搜索成功后, 在缓存表中存储此次资源搜索相关信息以备下次使用。缓存表采取 LRU (最近最少使用) 策略进行维护更新。

3.3 节点退出

节点的退出分为主动退出和被动退出。主动退出时, 节点 LP_i 会向所属的 SP_i 报告自己的退出消息, SP_i 会在本地资源数据库中删除 LP_i 所对应的节点及资源信息。此外,

LP_i 向自己的邻居节点通知即将离开, 邻居节点据此删除邻居表中 LP_i 的相关项, 同时, LP_i 还将向自己的前继节点和后继节点通知自己即将退出, 在 LP_i 的协助下, 进行如下操作:

$LP_i.successor.predecessor = LP_i.predecessor$;

$LP_i.predecessor.successor = LP_i.successor$

其他相关节点暂时不更新路由表。

被动退出时, 每个节点都会定时向其邻居节点发送心跳检测信息, 若在规定的时间内未收到邻居节点 LP_i 的响应, 则节点就认为 LP_i 已退出网络, 检测节点便向 SP_i 发送 LP_i 的退出消息, 并删除邻居表中 LP_i 的相关项。 SP_i 据此在本地资源数据库中删除 LP_i 的节点资源信息, 并协助 LP_i 的前继节点和后继节点完成连接, 而其他的节点则暂时不更新路由表。

3.4 拓扑维护(强稳定算法)

由于网络节点可以动态加入或退出, 为了保持网络的逻辑拓扑结构, SP_i 会周期性地对本地的全局逻辑拓扑维护操作, 即强稳定算法, 假定其周期为 T 。

当到达全局拓扑维护周期 T 时, SP_i 会根据本地资源数据库中的节点 ID 进行排序, 生成新的全局拓扑信息, 并向每个节点发送它所需要的双向节点信息, 节点会根据此信息更新自己的路由表。此外, 每个节点将更新自己的邻居表, 检测表中的所有邻居是否全部在线, 若通过检测, 在表中删除已退出网络的邻居节点的表项, 并为新邻居添加相应的表项。

4 改进的泛洪搜索机制

采用改进的泛洪搜索机制进行泛洪查找, 以布鲁曼滤波器^[5,6]这种随机数据结构进行搜索消息的转发, 且与迭代泛洪^[7]相结合, 并在搜索过程中通过计算语义相关度支持资源的模糊查找。将这种基于布鲁曼滤波器进行的泛洪搜索机制记为 BF_Flooding。

假设布鲁曼滤波器共有 m 位, 网络规模为 N , 为实现布鲁曼滤波器结构采用 k 个散列函数, 允许布鲁曼滤波器产生假阳性错误的最大值为 ε , 最小值为 f 。为了实现 f , 则 $k = \ln 2 \times (m/N)$; 而在散列函数的个数取到最优时, 要让错误率不超过 ε , 则 $m \geq N \times \ln e \times \ln(1/\varepsilon)$; 布鲁曼滤波器分别通过添加和查询操作判断添加成员和成员是否存在于布鲁曼滤波器中。

4.1 节点加入

节点 LP_i 加入网络的流程大体同第 3.1 节所述。此外,

LP_i 会初始化自己的布鲁曼滤波器, 假设在初始时刻 LP_i 未发出资源搜索请求且未收到来自其他节点的搜索请求, 即 LP_i 暂时不会转发查询消息, 此时, 将布鲁曼滤波器初始化为全零, 否则, 将要转发的目的邻居节点加入布鲁曼滤波器中。布鲁曼滤波器的结构如图 2 所示。其中, style 默认为泛洪消息报文; ID 表示报文发送者的全网 ID; message 是具体的消息内容。

0	1	...	0	$m-1$	ID	style	message
---	---	-----	---	-------	----	-------	---------

图 2 布鲁曼滤波器报文结构

4.2 查询路由

查询路由的具体流程如下。

(1) 当 LP_i 需要查询某个流媒体资源时, 节点首先查询本地缓存表, 看是否存有该关键字的查询记录, 若有, 则可直接定位到资源的源节点。

(2) 若缓存表中无相关记录, LP_i 向 SP_i 发出请求, SP_i 通过分析计算得出候选集, 若热门资源集中相关词条与输入的请求码字的平均语义相关度较大, 则 SP_i 指示 LP_i 采用改进的泛洪搜索机制, 即通过 BF_Flooding 和语义相关度计算相结合进行模糊查找。

(3) 若 LP_i 被指示采用改进的泛洪查找机制, LP_i 将所要搜索的资源关键字信息、迭代深度和语义相关度门限 φ 附入布鲁曼滤波器的 message 项中, 将迭代策略设置为 $P = \{a, b, c\}$ (a 为第一轮迭代深度; b 为第二轮迭代深度; c 为最大迭代深度)。 LP_i 按照一定的概率 f (本文中取最坏情况 $f=1$ 进行试验) 选择若干个邻居节点作为查询消息的目的转发节点, 并对这些节点进行添加操作, 将其加入布鲁曼滤波器中, 然后向这些目标转发节点以布鲁曼滤波器的结构发起 $TTL=a$ 的查询请求。

(4) 收到查询请求的节点 LP_j 将布鲁曼滤波器的 message 项中的资源关键字信息与本节点的资源列表进行比较, 找出相关项, 并按照式(1)计算其与请求关键字的语义相关度 λ , 若 $\lambda \geq \varphi$, 则将结果返回给 LP_i 。 LP_j 按照概率 f 选择若干个邻居节点作为本轮的目的转发节点, 继而对这些目的转发节点进行查询操作, 判断所选定的目的转发节点是否存在于所接收到的布鲁曼滤波器中(将此时收到的布鲁曼滤波器记为 receiver i , i 为当前的迭代深度), 若不存在, 则向这些目的转发节点发送查询消息, 否则, 不再向其发送查询消息。之后收到查询消息的节点都将重复 LP_j 的操作, 直到 $TTL=a$ 时暂停搜索。此时, 处在深



度为 a 的节点成为下一轮泛洪的前继节点, 前继节点保存 receiver a 消息, 并等待一定的时间 t 接收 LP_i 是否继续搜索的指示。

(5) 在完成深度为 a 的泛洪搜索后, 将结果按语义相关度降序排列后返回给 LP_i , LP_i 对返回结果进行处理。若满意该结果, 则搜索过程结束; 否则, LP_i 将启动下一轮的迭代, 进行深度为 b 的泛洪搜索。 LP_i 向存在于布鲁曼滤波器中的邻居节点发送第二轮迭代的启动消息 $renew$, 其 $TTL=b$ 。深度在 a 以内的节点只接收并转发这个消息, 并不处理。直到深度等于 a 的前继节点收到 $renew$ 后, 丢弃 $renew$ 并取出 receiver a 消息, 开始进行深度为 $(b-a)$ 的迭代搜索, 重复步骤(4)的过程, 直到 $TTL=c$ 时结束搜索。由于 $TTL=c$ 为最大的迭代深度, 泛洪搜索到此结束。

(6) 若 LP_i 不满意返回的结果, 则 LP_i 可激活以改进的 Chord 机制和 MKQ 结合进行模糊查找。当搜索完成后, 仍没有满意结果时, LP_i 将上报 SP, 由 SP 向上层 CS 请求资源。

(7) 同 DHT 搜索一样, LP_i 在搜索成功后, 将在缓存表中存储此次资源搜索相关信息以备下次使用。缓存表采取 LRU 策略进行维护更新。

4.3 节点退出

节点退出分为主动退出和被动退出, LP_i 的主动和被动退出流程大体同第 3.4 节所述, 只是其邻居节点在 LP_i 退出后, 还需将其布鲁曼滤波器中的 LP_i 信息删除。

5 算法仿真

实验中, 利用 BRUTE 产生不同节点规模 N 的底层 P2P 网络拓扑, 所有节点的资源存放策略和热门程度均服从 Zipf 分布。

5.1 改进的 Chord 搜索机制仿真

采用 DHT 搜索机制, 主要考察节点搜索过程中节点维护的路由表长度、查询消息转发次数和搜索成功率。传统的 Chord 算法的路由表长度为 $\lg N$, 改进的 Chord 算法的路由表长度为 $2 \times \lg N$ 。在不同的网络规模下, 每个节点所需要维护的路由表长度如图 3 所示, 在不同的网络规模下, 节点分别利用传统 Chord 算法和改进的 Chord 算法进行资源搜索时的查询消息转发, 转发次数对比如图 4 所示。由图 3 可知, 改进的 Chord 算法的路由表比传统的 Chord 算法的路由表长, 但增长较缓慢。随着网络规模的增大, 改进的 Chord 算

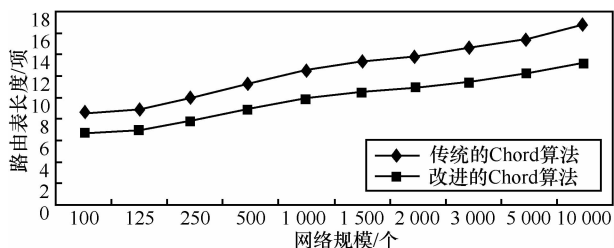


图3 改进的 Chord 算法与传统 Chord 算法路由表长度比较

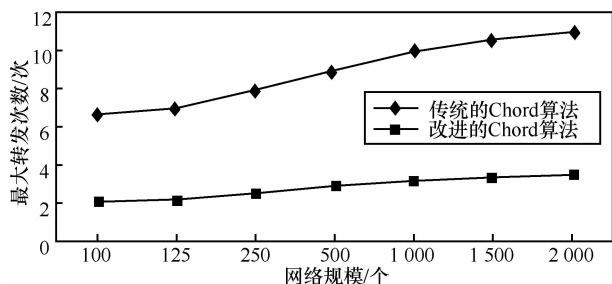


图4 改进的 Chord 算法与传统 Chord 算法最大转发次数比较

法的路由表长度增长并不明显, 但是路由消息转发次数却明显下降, 如图 4 所示。因此, 改进的 Chord 算法在网络规模较大时体现出明显的优越性。

采用 MKQ 支持模糊查找后, 改进的 Chord 算法的搜索成功率比较如图 5 所示, 由于引入了 MKQ, 改进的 Chord 算法只需在较小的转发次数内便可达到较高的查找成功率, 而传统 Chord 算法则需要将近 2 倍的转发次数才可达达到相同的查找成功率。

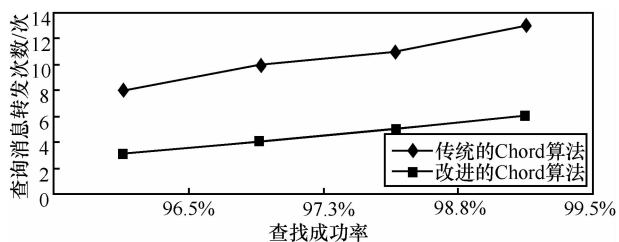
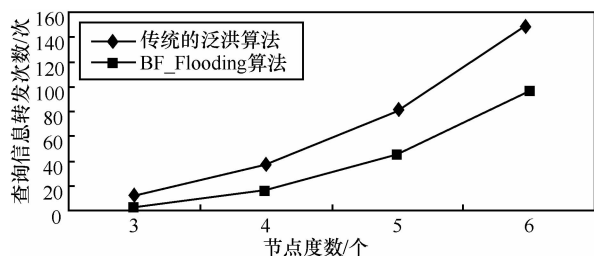
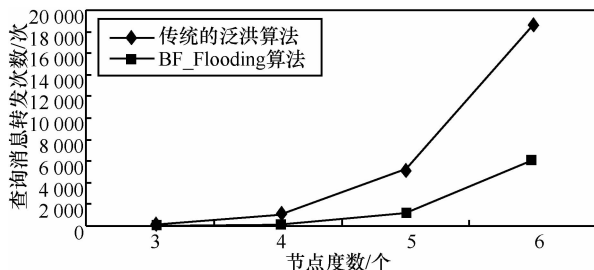


图5 不同 Chord 算法的搜索成功率

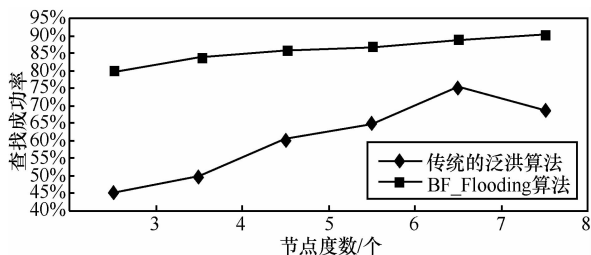
5.2 改进的泛洪搜索机制仿真

采用 BF_Flooding 搜索方式, 主要考察采用布鲁曼滤波器报文结构查询时, 网络中的冗余消息数量和搜索成功率。假设系统设定迭代最大深度为 q , 每个节点平均度数为 d 。在不同的迭代深度和邻居度数下, 查询消息在网络中的转发流量如图 6、图 7 所示。

由图 6、图 7 可知, 随着网络规模和节点度数的不断增加, 采用 BF_Flooding 算法在度数较大时, 其查询消息的转发量少于传统泛洪算法的一半, 从而大大减小网络和节点的负担。

图6 不同泛洪算法产生的转发流量($q=3$)比较图7 不同泛洪算法产生的转发流量($q=6$)比较

引入语义相关度匹配后的BF_Flooding算法与传统泛洪算法的搜索成功率的比较如图8所示。由图8可知, BF_Flooding算法通过引入语义相关度来支持模糊查找,提高了查询的命中率,其成功率稳定在80%左右,而传统泛洪算法远远低于该值。

图8 不同泛洪算法的搜索成功率($q=6$)比较

6 结束语

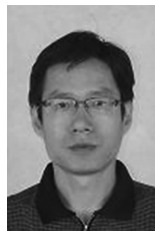
本文主要讨论了混合流媒体系统资源搜索机制,由于传统的搜索机制较单一,没有考虑到资源知名度对搜索算法的影响,本文采用改进的Chord机制和BF_Flooding结合的混合搜索方式进行资源查找,并且为了更好地提高搜索成功率,分别引入MKQ和语义相关度匹配来支持资源的模糊搜索。由仿真结果可知,与传统的Chord搜索算法相比,改进的Chord搜索算法在小幅度增加路由长度的代价下明显降低了查询路由时延,并且通过与MKQ结合,大大提高了搜索成功率。BF_Flooding算法与传统的泛洪算法相比,随着迭代深度和节点度数的增加,明显减少了

查询消息在网络中的重复传输次数,大大降低了查询消息在网络中引起的冗余流量,通过引入语义相关度,支持了资源的模糊搜索,提高了资源搜索的质量。

参考文献

- 1 施晓秋. 非集中式P2P系统中资源搜索与现存问题分析. 计算机工程, 2007, 33(5): 91~105
- 2 崔晓微, 董雷刚. 非结构化P2P搜索方法分析与展望. 大庆师范学院学报, 2011, 31(3): 13~16
- 3 王煜坤. 基于CDN和P2P技术的流媒体系统设计. 现代计算机, 2009(303): 179~181
- 4 刘震, 邓苏, 黄宏斌. 基于混合P2P网络模型的语义检索方法研究. 计算机科学, 2009, 36(12): 60~64
- 5 池静. Bloom Filter的研究和应用. 河北建筑科技学院学报, 2003, 20(4): 59~61
- 6 Bloom B. Space/time tradeoffs in hash coding with allowable errors. Communications of the ACM, 1970, 13(7): 422~426
- 7 张誉腾. 混合P2P网络信息搜索方法的应用研究. 湘潭大学硕士学位论文, 2011

[作者简介]



郭大钢,男,西安邮电学院邮电技术公司工程师,主要研究方向为移动通信。



卓明琴,女,中兴通讯股份有限公司南京研发中心工程师,主要研究方向为通信软件开发。



张继荣,女,西安邮电大学通信与信息工程学院教授,主要研究方向为宽带通信网。

(收稿日期:2013-12-15)